Scala编译器是如何解析for循环语句

你可能会在Scala中经常使用for循环已经，所以理解Scala编译器是如何解析for循环语句是非常重要的。我们记住以下四点规则即可：

1、对集合进行简单的for操作，Scala编译器会将它翻译成对集合进行foreach操作；

2、带有guard的for循环，编译器会将它翻译成一序列的withFilter操作，紧接着是foreach操作；

3、带有yield的for循环，编译器会将它翻译成map操作；

4、带有yield和guard的for循环，编译器会将它翻译成一序列的withFilter操作，紧接着是map操作。

光说没有说服力，我们来举例子一一说明吧。

一、对集合进行简单的for操作，Scala编译器会将它翻译成对集合进行foreach操作

我们有如下的代码：

```
class Iteblog{
   for (i <- 1 to 10) println(i)
}
```

我们可以使用scalac -Xprint:parse Itrblog.scala来查看编译器是如何翻译上面的代码的：

```
[iteblog@www.iteblog.com scala]$ scalac -Xprint:parse Itrblog.scala
[[syntax trees at end of                     parser]] // Itrblog.scala
package <empty> {
  class Iteblog extends scala.AnyRef {
    def <init>() = {
      super.<init>();
      ()
    };
    1.to(10).foreach(((i) => println(i)))
  }
}
```

看到1.to(10).foreach(((i) => println(i)))这行代码吗？他就是编译器将for (i

二、带有guard的for循环，编译器会将它翻译成一序列的withFilter操作，紧接着是foreach操作

```
class Iteblog2{
  for {
    i <- 1 to 10
    if i % 2 == 0
  } println(i)
}
```

编译后变成的结果：

```
[iteblog@www.iteblog.com scala]$ scalac -Xprint:parse Iteblog1.scala
[[syntax trees at end of                     parser]] // Iteblog1.scala
package <empty> {
  class Iteblog2 extends scala.AnyRef {
    def <init>() = {
      super.<init>();
      ()
    };
    1.to(10).withFilter(((i) => i.$percent(2).$eq$eq(0))).foreach(((i) => println(i)))
  }
}
```

三、带有yield的for循环，编译器会将它翻译成map操作

```
class Iteblog2{
  for { i <- 1 to 10 } yield i
}
```

编译后变成的结果：

```
[iteblog@www.iteblog.com scala]$ scalac -Xprint:parse Iteblog2.scala
[[syntax trees at end of                     parser]] // Iteblog2.scala
package <empty> {
```

```
 class Iteblog2 extends scala.AnyRef {
   def <init>() = {
     super.<init>();
     ()
   };
   1.to(10).map(((i) => i))
 }
}
```

四、带有yield和guard的for循环，编译器会将它翻译成一序列的withFilter操作，紧接着是map操作

```
class Iteblog3{
 for {
   i <- 1 to 10
   if i % 2 == 0
 } yield i
}
```

编译后变成的结果：

```
[iteblog@www.iteblog.com scala]$ scalac -Xprint:parse Iteblog3.scala
[[syntax trees at end of                     parser]] // Iteblog3.scala
package <empty> {
 class Iteblog3 extends scala.AnyRef {
   def <init>() = {
     super.<init>();
     ()
   };
   1.to(10).withFilter(((i) => i.$percent(2).$eq$eq(0))).map(((i) => i))
 }
}
```

本文链接: 【】（）